COMPUTER SCIENCE TRIPOS Part II

Tuesday 1 June 1999 1.30 to 4.30

Paper 7

Answer **five** *questions.*

Submit the answers in five **separate** *bundles, each with its own cover sheet. On each cover sheet, write the numbers of* **all** *attempted questions, and circle the number of the question attached.*

Write on **one** *side of the paper only.*

1 VLSI

Describe how a collection of boolean functions expressed as the sum of products of a given set of input signals can be implemented as a programmable logic array in nMOS. Illustrate your example by showing the stick diagram layout for the following pair of functions:

$$F = A.B + B'.C + C'.A'$$
$$G = C.A + A'.B + B'.C'$$

[It is not necessary to expand the full detail of any inverters used in the circuit.]

[12 marks]

Explain how the depletion mode pull-up transistors might be replaced in CMOS by

(*a*) p-channel transistors used as passive pull-ups; [4 marks]

(*b*) dynamic logic based on NORA. [4 marks]

[It is not necessary to give the detailed stick diagram for these.]

2 Specification and Verification I

For $n \geqslant 0$, define $\Sigma(A, n)$ recursively by:

$$\Sigma(A,n) \;=\; \textit{if } n=0 \textit{ then } A(0) \textit{ else } \Sigma(A,n{-}1)+A(n)$$

For $m \geqslant 0$ and $n \geqslant 0$ define $\Sigma2(A, m, n)$ by:

$$\begin{array}{ll}\Sigma2(A,m,n) = & \textit{if } m=0 \\ & \textit{then } \Sigma(A,n) \\ & \textit{else} \\ & \;\textit{if } m \leqslant n \textit{ then } \Sigma(A,n)-\Sigma(A,m{-}1) \textit{ else } 0\end{array}$$

Show that:

(*a*) $\forall n.\ 0 \leqslant n \;\Rightarrow\; \Sigma2(A,n,n) = A(n)$ [2 marks]

(*b*) $\forall m\ n.\ 0 \leqslant m \wedge m < n \;\Rightarrow\; \Sigma2(A,m,n) = A(m)+A(n)+\Sigma2(A,m{+}1,n{-}1)$ [6 marks]

Give a detailed proof using Floyd–Hoare logic that:

$\vdash \;\{\mathtt{N} = n \;\wedge\; n \geqslant 0\}$

```
M := 0; SUM := 0;
WHILE M ⩽ N DO
  IF M=N
    THEN BEGIN SUM := SUM + A(N); M := M+1 END
    ELSE BEGIN SUM := SUM+A(M)+A(N); M := M+1; N := N-1 END
```

$\{\mathtt{SUM} = \Sigma(\mathtt{A}, n)\}$

[12 marks]

3 Comparative Architectures

Why might a processor supporting "dynamic execution" (out-of-order execution with speculation) yield better performance than a statically-scheduled processor with a similar number of execution units? [5 marks]

Outline the operation of a dynamic execution processor. You should address the following issues in your answer:

- register re-use (name dependencies)
- arithmetic exceptions
- control-flow misprediction
- load/store ordering preservation

[10 marks]

What factors are likely to affect the amount of instruction-level parallelism that will be exploited by future processors? [5 marks]

4 Optimising Compilers

Define the notion of a program expressed as a flowgraph of three-address instructions being in *single static assignment* (SSA) form. [4 marks]

Given a program expressed in a language like C, explain how one might deal with

(*a*) temporaries used to express a complicated expression as a sequence of three-address instructions, and

(*b*) (non-address taken) formal parameters and local variables

so that the resulting flowgraph is in SSA form. [6 marks]

Give a program for which performing a "map to SSA form" pass before register allocation is likely to result in better code, noting any assumptions you make on how register allocation works. [5 marks]

Commonly SSA form is discussed only for temporaries and non-address-taken local variables. To what extent is it possible to arrange that accesses to address-taken locals or statically allocated global variables can be transformed into SSA form? What effect might such a transformation have on register allocation of such variables? [5 marks]

5 Information Retrieval

You are asked to design a modern information management system for a large multi-volume encyclopedia like the *Encyclopaedia Britannica*.

Describe in detail the ways in which you would organise and index the encyclopedia text, and the facilities you would offer the system user. [12 marks]

Justify your choices. [4 marks]

Comment on the issues that would arise if you wanted to evaluate system performance. [4 marks]

6 Security

You have been hired by a company which is bidding to take over the National Lottery when Camelot's franchise expires, and your responsibility is the security architecture. State the security policy you would recommend and outline the mechanisms you would implement to enforce it. [20 marks]

7 Neural Computing

Explain the mechanisms and computational significance of nerve impulse generation and transmission. Discuss each of the following aspects:

(*a*) The equivalent electrical circuit for a nerve cell membrane.

(*b*) How different ion species flow across the membrane, in terms of currents, capacitance, conductances, and voltage-dependence. (Your answer can be qualitative.)

(*c*) The role of positive feedback and voltage-dependent conductances.

(*d*) The respect in which a nerve impulse is a mathematical catastrophe.

(*e*) The approximate time-scale of events, and the speed of nerve impulse propagation.

(*f*) What happens when a propagating nerve impulse reaches an axonal branch.

(*g*) What would happen if two impulses approached each other from opposite directions along a single nerve fibre and collided.

(*h*) How linear operations like integration in space and time can be combined in dendritic trees with logical or boolean operations such as AND, OR, NOT, and veto.

(*i*) Whether "processing" can be distinguished from "communications", as it is for artificial computing devices.

(*j*) The respects in which stochasticity in nerve impulse time-series may offer computational opportunities that are absent in synchronous deterministic logic.

[2 marks each]

8 Natural Language Processing

Write about *two* of the following. Describe how each technique can be used in the solution of a natural language processing (sub)task and the problems which remain.

(*a*) Active chart parsing

(*b*) Part-of-speech disambiguation using finite-state machines

(*c*) Probabilistic context-free grammar

(*d*) Plan recognition for discourse comprehension

[10 marks each]

9 Denotational Semantics

Suppose that $f : D \to D$ is a continuous function on a domain. What is meant by the *least pre-fixed point*, $\mathit{fix}(f)$, of f? [2 marks]

Show that $\mathit{fix}(f)$ exists and is in fact the least fixed point of f. [12 marks]

Suppose now that E is another domain and $g : D \times E \to E$ a continuous function. Let (d, e) be the least element of $D \times E$ satisfying

$$\begin{cases} d &= f(d) \\ e &= g(d, e) \end{cases}$$

Prove that $d = \mathit{fix}(f)$. [6 marks]

10 Specification and Verification II

Consider the following Verilog phrases:

```
initial r = 0;
always @(posedge clk) r = a + r;
```

Write down a formula in logic that relates `clk`, `a` and `r` at a level of abstraction where clock edges are explicitly represented. [4 marks]

Write down a second formula that models only the sequences of values of `a` and `r` at successive clock cycles. [4 marks]

Discuss the relationship between the two formulae. [4 marks]

Formalise and prove, using your second formula, that on the n^{th} cycle the value of `r` is the sum of the values of `a` on all the cycles up to the n^{th}. You may assume that values are natural numbers and ignore the possibility of overflow. [8 marks]

11 Advanced Graphics

Give a parametric definition of a torus centred at the origin and aligned with the coordinate axes. [4 marks]

Outline how you would find the first intersection point, if any, of a ray with the torus from the previous part. [You may assume that you are provided with functions to find the roots of linear, quadratic, cubic and quartic equations.] [4 marks]

Show how to construct a circle using non-uniform rational B-splines (NURBS). [8 marks]

Show how the circle definition from the previous part can be used to define a NURBS torus. [You need explain only the general principle and the location of the torus's control points.] [4 marks]

12 Types

Contrast the advantages and disadvantages of explicit and implicit typing. [4 marks]

Consider an ML-style language with types and type schemes

$$\begin{aligned} \tau &::= \alpha \mid \texttt{bool} \mid \tau \to \tau \mid \tau\ \texttt{list} \\ \sigma &::= \forall A\,(\tau) \end{aligned}$$

Give the typing rules for variables, function abstraction, function application, and let-binding. Make the form of the typing judgement clear. [5 marks]

Give terms M_1 and M_2 such that N_1 is typable (in the empty context) and N_2 is not, where

$$\begin{aligned} N_1 &\stackrel{\text{def}}{=} \textbf{let}\,\textbf{val}\, f = M_1\, \textbf{in}\, M_2\, \textbf{end} \\ N_2 &\stackrel{\text{def}}{=} (\textbf{fn}\ f \Rightarrow M_2)\, M_1 \end{aligned}$$

Give all uses of $\sigma \succ \tau$ required in a typing derivation for N_1; prove that there does not exist a typing derivation for N_2. [7 marks]

What is a *principal type scheme*? Give the principal type scheme for N_1, or explain informally why it does not have one. [4 marks]

13 Communicating Automata and Pi Calculus

Concurrent processes are defined by the syntax

$$P \quad ::= \quad A\langle b_1, \ldots, b_n \rangle \;\Big|\; \Sigma \alpha_i.P_i \;\Big|\; P_1 \mid P_2 \;\Big|\; \mathsf{new}\, a\, P$$

where each process identifier A is equipped with a defining equation $A(a_1, \ldots, a_n) \stackrel{\text{def}}{=} P_A$. Give the transition rules from which transitions of the form $P \stackrel{\alpha}{\to} P'$ can be inferred, where α is of the form a, $\overline{a}$ or τ. The rules should not use structural congruence ($\equiv$). [5 marks]

Enumerate the ways in which a transition of the form $P|Q \stackrel{\alpha}{\to} R$ can be inferred from transitions of P and/or Q, and indicate the form of R in each case. [5 marks]

Hence show that if $P|Q \stackrel{\alpha}{\to} R_1$, then there exists R_2 such that $Q|P \stackrel{\alpha}{\to} R_2$ and $R_1 \equiv R_2$. [5 marks]

Give an example of P and Q for which $\mathsf{new}\, a(P|Q)$ has a τ-transition but $P|\mathsf{new}\, a\, Q$ has no τ-transition. Now suppose that $\mathsf{new}\, a(P|Q) \stackrel{\alpha}{\to} R_1$; what syntactic condition on P ensures that $P|\mathsf{new}\, a\, Q \stackrel{\alpha}{\to} R_2$ for some R_2 with $R_1 \equiv R_2$? Justify your answer. [5 marks]

14 Additional Topics

List some intrinsic properties of wearable computers and how they may develop in future. [6 marks]

What is an Active Badge and how can it be used to locate personnel and equipment? [6 marks]

Give *four* other applications of the Active Badge and write brief notes on each. [4 marks]

Choose *one* of these four applications and explain how the underlying distributed system can ensure that the appropriate quality of service is available to the user. [4 marks]

15 Additional Topics

Justify each of the following statements, with the help of some formal details and examples.

"The dynamic behaviour of programs in a programming language can be defined in terms of a formal logical system, in which each sentence which can be inferred in the logic represents the evaluation of a piece of program."

"The type discipline of a programming language can be defined in terms of a formal logical system, in which each sentence which can be inferred in the logic represents the assertion that a piece of program obeys the discipline."

[15 marks]

Give and explain *two* examples of theorems about the behaviour of Standard ML programs which can be expressed and proved in terms of either or both of the two logical systems described above. [5 marks]